IN THE CLAIMS

1. (Previously Presented) In a computer environment where devices are occasionally connected together, a method for automated transmission and execution of an executable file of interest originating from a first device, upon the first device's connection to a second device, the method comprising:

connecting the first device to at least one other device capable of hosting the first device;

identifying at least one particular host device that is connected to the first device, including determining communication information allowing communication between the first device and the particular host device, and determining command information allowing the first device to invoke execution of the application or driver of interest at the particular host device;

based on said determined communication information, transmitting the executable file of interest from said first device to the particular host device; and

based on said determined command information, invoking execution of the executable file of interest after it has been transmitted to the particular host device; and

entering a listening mode, awaiting commands from the executable file running on the particular host device.

2. (Previously presented) The method of claim 1, wherein said executable file of interest comprises a device driver file.

3. (Previously presented) The method of claim 2, wherein said device driver file, upon execution, controls the interaction between the first device and the particular host device, and further controls the operation of said first device.

4. (Original) The method of claim 1, wherein said executable file comprises a binary file having instructions capable of executing at said second device.

5. (Original) The method of claim 1, wherein said executable file comprises an application program capable of executing at said second device.

6. (Original) The method of claim 1, wherein said first device includes an add-in device capable of being hosted by said second device.

7. (Original) The method of claim 6, wherein said first device comprises a digital camera device and wherein said method further comprises:

upon execution of said executable file at said second device, transferring image information from said digital camera device to said second device.

8. (Original) At the method of claim 7, further comprising:

after transferring said image information from said digital camera device to said second device, wirelessly transmitting said image information to a third device.

9. (Original) The method of claim 1, wherein said second device includes a computing device capable of hosting other devices.

10. (Original) The method of claim 1, wherein said second device includes wireless transmission capability for transferring information received from said first device to other devices.

11. (Original) The method of claim 1, wherein said first and second devices are occasionally connected together.

12. (Original) The method of claim 1, wherein said first and second devices are permanently connected together.

13. (Previously presented) The method of claim 1, wherein said first and second devices are connected together via one or more of the following: a serial communication link, an RS-232 serial communication link, a universal serial bus (USB) link.

14-15. (Canceled)

16. (Original) The method of claim 1, wherein invocation of said identifying step occurs upon connecting said first and second devices together.

17. (Original) The method of claim 1, wherein said identifying step includes:

probing the first device's environment for determining which devices, if any, the first device is attached to.

18. (Original) The method of claim 17, wherein said probing step includes:

determining a default communication medium for probing for new devices.

19. (Original) The method of claim 18, wherein said default communication medium is specified initially by factory-preset information.

20. (Original) The method of claim 18, wherein said default communication medium is a selected one of wireless and wired communication medium.

21. (Original) The method of claim 20, wherein said default communication medium includes serial (RS-232) and USB (Universal Serial Bus) wired communication medium.

22. (Original) The method of claim 19, wherein said factory-preset information is stored in a registry of the first device.

23. (Original) The method of claim 19, wherein said factory-preset information includes a default communication rate and default handshake protocol for at least one potential host device.

24. (Original) The method of claim 17, wherein said probing step includes:

executing an initial sequence of handshake commands and comparing any response received to a list of known responses for identifying a particular host device.

25. (Original) The method of claim 17, wherein said probing step continues until all known potential host devices have been enumerated.

26. (Original) The method of claim 1, wherein said identifying step includes:

updating a registry at said first device for indicating any connected host device that has been identified.

27. (Original) The method of claim 1, further comprising:

upon identifying at least one particular host device, ensuring that a state of TCP/IP communication is reached between said first device and the particular identified host device.

28. (Original) The method of claim 27, wherein said step of ensuring that a state of TCP/IP communication is reached includes:

initiating a PPP (Point-to-Point Protocol) communication session between said first and second devices, and, thereafter

initiating a TCP/IP communication session between said first and second devices.

29. (Original) The method of claim 27, wherein said step of ensuring that a state of TCP/IP communication is reached includes:

determining an IP (Internet Protocol) address for said second device.

30. (Original) The method of claim 1, wherein said step of transmitting the executable file of interest includes:

opening the executable file of interest at the first device; and

streaming the opened executable file of interest from the first device to the second device.

31. (Currently Amended) The method of claim 30, wherein said streaming step includes:

employing Extensible Markup Language (XML) file format ~~protocol~~ for packaging said executable file of interest for delivery to the second device.

32. (Original) The method of claim 30, wherein said step of transmitting further comprises:

returning to said first device a file handle permitting said first device to access said executable file of interest transmitted to said second device.

33. (Original) The method of claim 31, wherein said file handle comprises a file handle that may be understood by said second device for accessing a particular file of interest at said second device.

34. (Original) The method of claim 1, wherein said executable file of interest comprises a byte-code program, and wherein said second device includes capability for executing byte-code programs.

35. (Original) The method of claim 1, wherein said executable file of interest comprises a Java program, and wherein said second device includes a Java Virtual Machine for executing Java programs.

36. (Original) The method of claim 1, wherein said step of invoking execution of the executable file of interest includes:

issuing a command from said first device to said second device to begin execution at said second device of said executable file of interest.

37. (Original) The method of claim 1, wherein said step of invoking execution of the executable file of interest includes:

triggering execution of said executable file indirectly at said second device by instructing said second device to restart itself.

38. (Original) The method of claim 1, further comprising:

placing said first device in a listening mode, after said first device has invoked execution of said executable file at said second device.

39. (Original) The method of claim 38, wherein said first device awaits commands from said second device, while said first device is in a listening mode.

40. (Original) The method of claim 39, wherein commands received at said first device from said second device control operation of said first device.

41. (Previously Presented) A multi-device system providing automated loading and execution of a device driver required for enabling interaction between connected devices, the system comprising:

a first device that may be connected to a second device that is capable of hosting the first device; and

a subsystem, incorporated in the first device, for automatically:

identifying the second device upon connection to the first device, said subsystem initiating communication between the two devices;

uploading the device driver of interest from the first device to the second device; and

transmitting at least one command from the first device that invokes execution of the driver of interest at the second device, whereupon the driver executes at the second device, the driving for controlling the interaction between the first device and the second device, and further for controlling the operation of the first device.

42. (Previously presented) The system of claim 41, wherein said device driver comprises a binary file having instructions capable of executing at said second device.

43. (Original) The system of claim 42, wherein said binary file comprises native machine instructions for execution by a processor at said second device.

44. (Original) The system of claim 42, wherein said binary file comprises byte-code instructions for execution by an interpreter at said second device.

45. (Original) The system of claim 44, wherein said binary file comprises a Java program and wherein said second device includes a Java Virtual Machine for executing Java programs.

46. (Original) The system of claim 44, wherein said driver includes:

instructions for unpacking other executable files for execution at said second device.

47. (Original) The system of claim 41, wherein said first device comprises an add-in device capable of being hosted by said second device.

48. (Original) The system of claim 47, wherein said first device comprises a digital camera device, and wherein said second device comprises a handheld device capable of hosting said digital camera device.

49. (Original) The system of claim 48, wherein said handheld computing device functions to retrieve digital image information from said digital camera device and wirelessly transmit that information to another system.

50. (Original) The system of claim 48, wherein said handheld device is a selected one of a cellular phone device and a handheld computing device.

51. (Previously Presented) A client device comprising:

a physical manager identify a host coupled to the client device;

a TCP/IP stack to initiate a communication session with the host;

an application/driver uploader to upload an object of interest onto the host device, the object of interest determined based on the identity of the host device determined by the physical manager; and

a file handle returned to the client device by the host device to allow the client device to perform a variety of operations on the uploaded object of interest as it resides at the host device, including starting an application or driver in the object of interest.

52. (Previously Presented) The client device of claim 51, further comprising: a registry to serve as a repository indicating various configuration settings, such as TCP/IP configuration settings, that are set in order to communicate with a particular host device.